# 

# Middlesex County College

**Edison, NJ 08818**

**COURSE ID:** Computer Science and Information Technology Department - CSC 267

# COURSE NAME: Advanced C#.NET Programming

### NUMBER OF CREDITS AWARDED FOR COURSE: 4 credits

**PREREQUISITE OR COREQUISITE COURSES OR ACADEMIC STANDING:**

Prerequisite(s): CSC 266 - C#.NET Programming

### NEW OR MODIFIED COURSE: New

# SEMESTER AND YEAR COURSE WILL FIRST BE OFFERED: Spring 2013

#### NAME AND TELEPHONE NUMBER OR EMAIL ADDRESS OF DEPARTMENT CHAIR:

Chairperson or Course Coordinator: Dr. Peter Farrett

Office Location: JLC201

E-mail Address: PFarrett@Middlesexcc.edu

Telephone: 732-906-2526

**DETAILED COURSE DESCRIPTION:**

This course investigates the software engineering principles of encapsulation, information hiding and code reuse, and discusses how these concepts are used to build abstract data types. The object oriented programming features of classes, inheritance, polymorphism and composition are studied, along with constructors and method overloading. Students are also exposed to the C#.NET programming concepts of exception handling, collections, and generics. In addition, students will query in-memory data by using LINQ and integrate code written in other languages into a C# application. This course is not designed for transfer.

# OUTLINE OF COURSE OBJECTIVES:

1. Manage the lifetime of objects and control the use of resources.
2. Catch, handle and throw exceptions.
3. Use inheritance to create new reference types.
4. Define properties and indexers to encapsulate data, and define operators for this data.
5. Decouple an operation from the method that implements an operation, and use these decoupled operations to handle asynchronous events.
6. Use collections to aggregate data, and use Generics to implement type-safe collection classes, structures, interfaces, and methods.
7. Implement custom collection classes that support enumeration.
8. Query in-memory data by using LINQ.
9. Integrate C# code with dynamic language code (Ruby and Python) and COM (Common Object Model) components.
10. Demonstrate how to use C# and Microsoft Visual Studio to build advanced .NET Framework applications.

**TEXTS, JOURNALS, AND OTHER MATERIALS USED IN COURSE:**

Doyle, Barbara. C# Programming 3E, From Problem Analysis to Program Design. Boston: Course Technology, Cengage Learning, 2011.

ISBN: 978-0-538-45302-8

## SUGGESTED GRADING CRITERIA:

3 Examinations 30%

Final Examination 20%

Lab Assignments 25%

Quizzes 10%

Homework 15%

100%

**SCHEDULE OF TOPICS TO BE COVERED:**

1. Managing the Lifetime of Objects and Controlling Resources

Introduction to Garbage Collection

Managing Resources

2. Handling Exceptions

Handling Exceptions

Raising Exceptions

3. Inheriting From Classes and Implementing Interfaces

Using Inheritance to Define New Reference Types

Defining and Implementing Interfaces

Defining Abstract Classes

4. Encapsulating Data and Defining Overloaded Operators

Creating and Using Properties

Creating and Using Indexers

Overloading Operators

5. Decoupling Methods and Handling Events

Declaring and Using Delegates

Using Lambda Expressions

Handling Events

6. Using Collections and Building Generic Types

Using Collections

Creating and Using Generic Types

Defining Generic Interfaces and Understanding Variance

Using Generic Methods and Delegates

7. Building and Enumerating Custom Collection Classes

Implementing a Custom Collection Class

Adding an Enumerator to a Custom Collection Class

8. Using LINQ to Query Data

Using the LINQ Extension Methods and Query Operators

Building Dynamic LINQ Queries and Expressions

9. Integrating Visual C# Code with Dynamic Languages and COM Components

Integrating C# Code with Ruby and Python

Accessing COM Components from C#

**Course Abstract**

**Course ID and Name:** CSC 267 – Advanced C#.NET Programming

**Department:** Computer Science and Information Technology

Chairperson or Course Coordinator: Dr. Peter Farrett

Office Location: JLC201

E-mail Address: PFarrett@Middlesexcc.edu

Telephone: 732-906-2526

**Prerequisites:** CSC 266, C#.NET Programming **Co-requisites:** None

**Course Description:** This course investigates the software engineering principles of encapsulation, information hiding and code reuse, and discusses how these concepts are used to build abstract data types. The object oriented programming features of classes, inheritance, polymorphism and composition are studied, along with constructors and method overloading. Students are also exposed to the C#.NET programming concepts of exception handling, collections, and generics. In addition, students will query in-memory data by using LINQ and integrate code written in other languages into a C# application. This course is not designed for transfer.

**General Education Status:** No

**Credits:** 4 **Lecture Hours:** 2  **Lab Hours:** 4

**Learning Outcomes:**

1. Manage the lifetime of objects and control the use of resources.
2. Catch, handle and throw exceptions.
3. Use inheritance to create new reference types.
4. Define properties and indexers to encapsulate data, and define operators for this data.
5. Use collections to aggregate data, and use Generics to implement type-safe collection classes, structures, interfaces, and methods.
6. Query in-memory data by using LINQ.

**Course Content Areas:**

1. Managing the Lifetime of Objects and Controlling Resources

Introduction to Garbage Collection

Managing Resources

1. Handling Exceptions

Handling Exceptions

Raising Exceptions

1. Inheriting From Classes and Implementing Interfaces

Using Inheritance to Define New Reference Types

Defining and Implementing Interfaces

Defining Abstract Classes

1. Encapsulating Data and Defining Overloaded Operators

Creating and Using Properties

Creating and Using Indexers

Overloading Operators

1. Decoupling Methods and Handling Events

Declaring and Using Delegates

Using Lambda Expressions

Handling Events

1. Using Collections and Building Generic Types

Using Collections

Creating and Using Generic Types

Defining Generic Interfaces and Understanding Variance

Using Generic Methods and Delegates

1. Building and Enumerating Custom Collection Classes

Implementing a Custom Collection Class

Adding an Enumerator to a Custom Collection Class

1. Using LINQ to Query Data

Using the LINQ Extension Methods and Query Operators

Building Dynamic LINQ Queries and Expressions

1. Integrating Visual C# Code with Dynamic Languages and COM Components

Integrating C# Code with Ruby and Python

Accessing COM Components from C#